



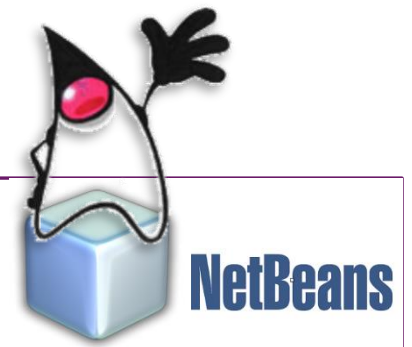
# TECNOLOGIA EM DESENVOLVIMENTO DE SOFTWARE MULTIPLATAFORMA

TÉCNICA DE PROGRAMAÇÃO I



TÉCNICA DE PROGRAMAÇÃO I  
PROFº LUIZ CLÁUDIO

# ESTRUTURA BÁSICA DE UMA CLASSE DE MODELAGEM



Atributo

Construtores

Getters e setters

Métodos específicos da classe

# NETBEANS – COMPONENTES UTILIZADOS



Abaixo são mostrados alguns dos componentes mais usados:

❑ **JLabel** - Exibe texto não editável (LABEL)

❑ **TextField** – Caixa de Texto

❑ **Button** – Botão - Libera um evento quando o usuário clicar nele com o mouse.

NETBEANS -



NetBeans



## Interface gráfica com Usuário – GUI(Graphical User Interface)

É onde os resultados são apresentados em modo gráfico. Essa interface é formada através de componentes GUI, conhecidos por **controles** ou **widgets**. Esses componentes são objetos que fazem a interação com usuário por teclado, mouse ou outros dispositivos que venham a servir para entrada de dados.

# NETBEANS



# NetBeans



Os componentes GUI Swing estão dentro do pacote **javax.swing** que são utilizados para construir as interfaces gráficas. Alguns componentes não são do tipo GUI Swing e sim componentes **AWT**. Antes de existir o GUI Swing, o Java tinha componentes **AWT** (Abstract Windows Toolkit) que faz parte do pacote **javax.awt**.

## Controles Swing

Label Rótulo

OK Button

ON Botão Alternar

Caixa de seleção

Caixa de combinação

Lista

## AWT

A Rótulo

ab Campo de texto

Caixa de seleção

Lista

Painel de rolagem

Canvas

Menu pop-up

OK Button

Área de texto

Opções

Barra de rolagem

Painel

Barra de menu

NETBEANS -



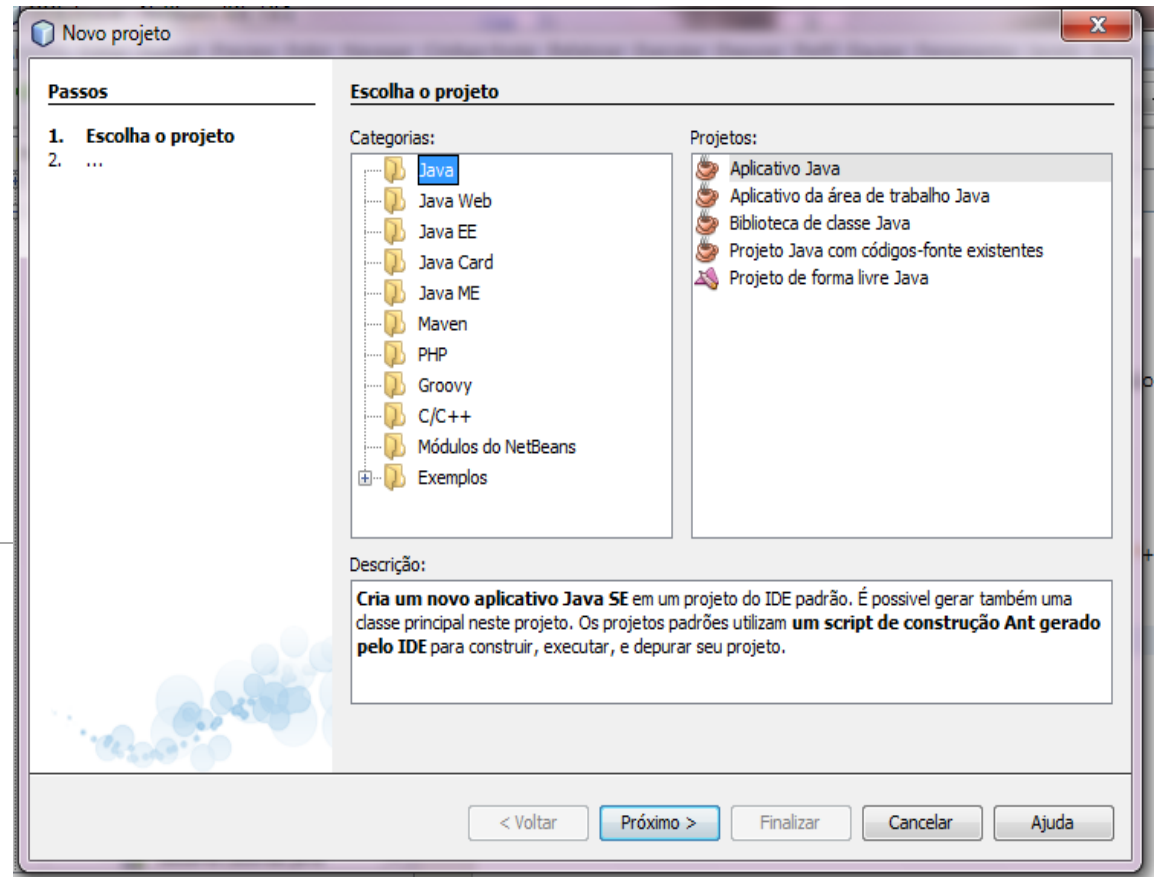
NetBeans



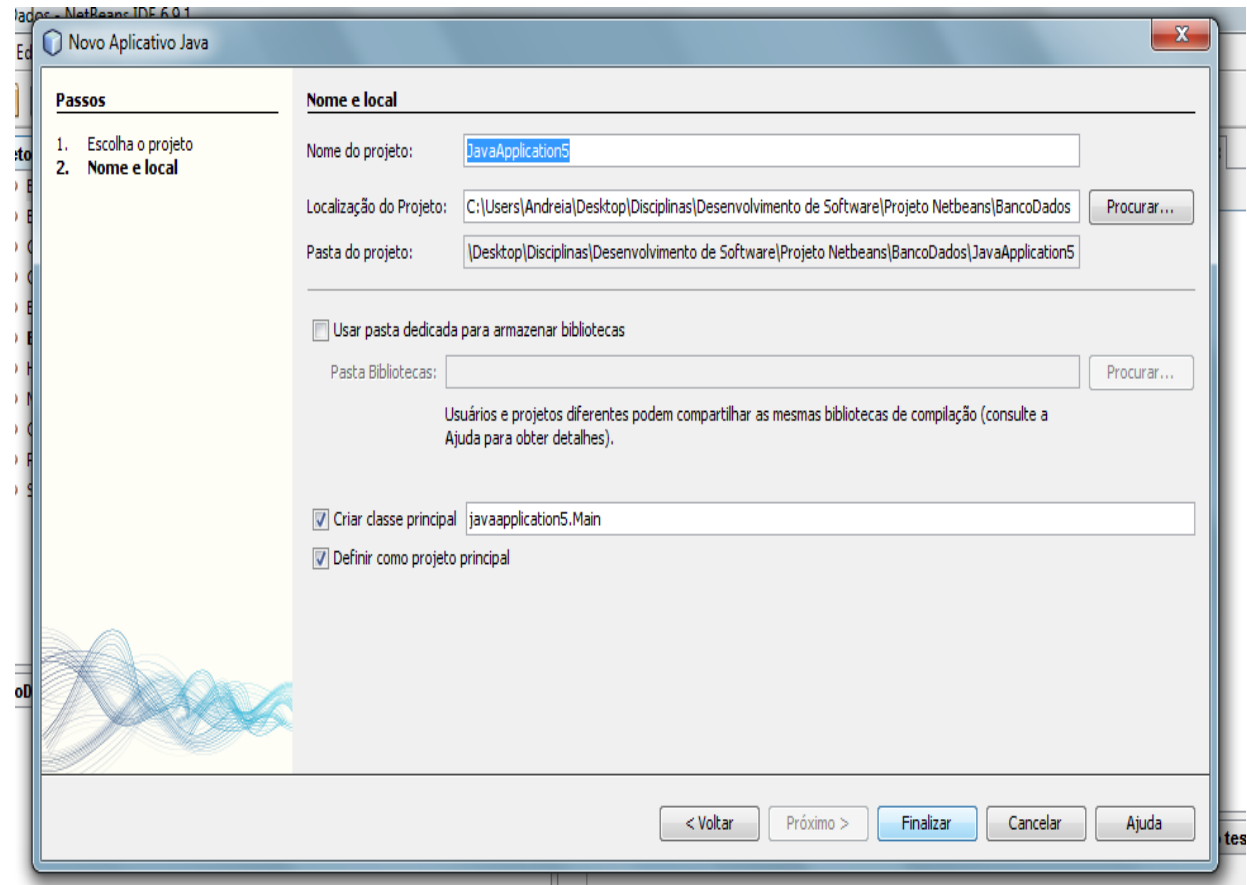
A diferença entre o GUI Swing e AWT, é na aparência e comportamento dos componentes, ou seja, quando criado por AWT, a aparência e comportamento de seus componentes são diferentes para cada plataforma e enquanto feito por GUI Swing, a aparência e comportamento funcionam da mesma forma para todas as plataformas. Os componentes AWT são mais pesados, pois requerem uma interação direta com o sistema de janela local, podendo restringir na aparência e funcionalidade, ficando menos flexíveis do que os componentes GUI Swing.

# Criar um projeto no NetBeans

1. CLICAR EM ARQUIVO -> NOVO PROJETO
2. JAVA – APLICATIVO JAVA
3. CLICAR EM PRÓXIMO



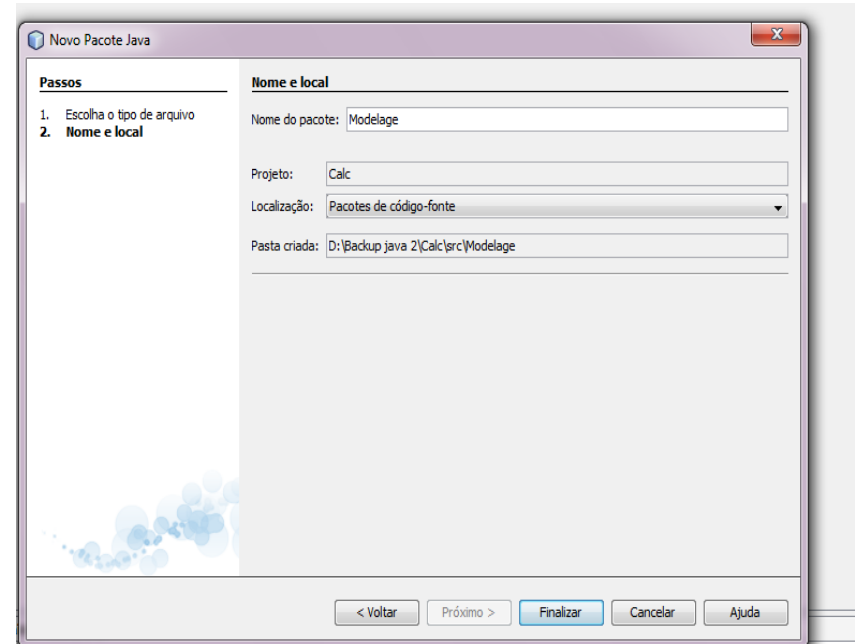
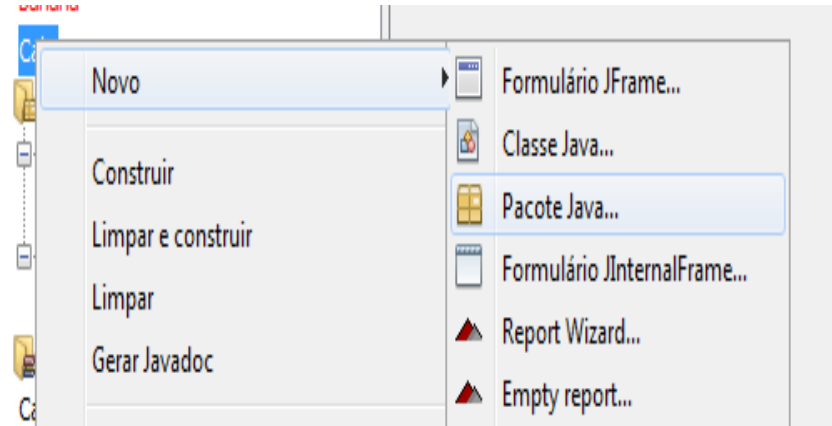
4. Digite o nome do projeto
5. Clicar em Finalizar





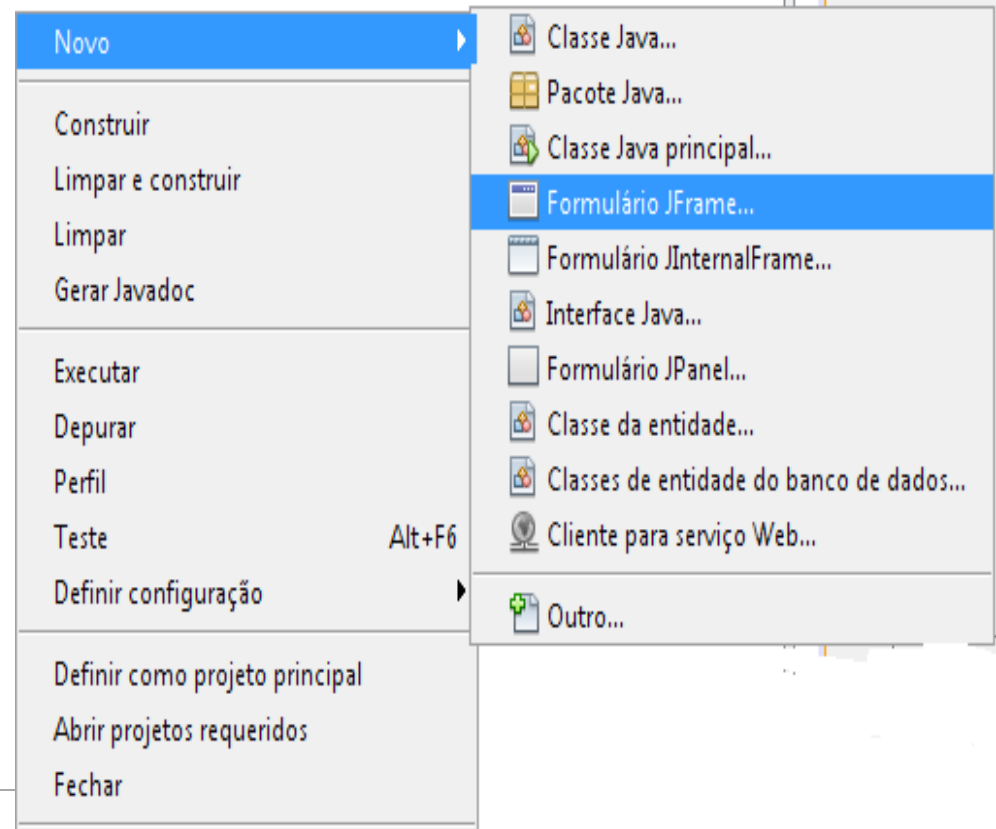
Para criar pacotes faça os passos a seguir:

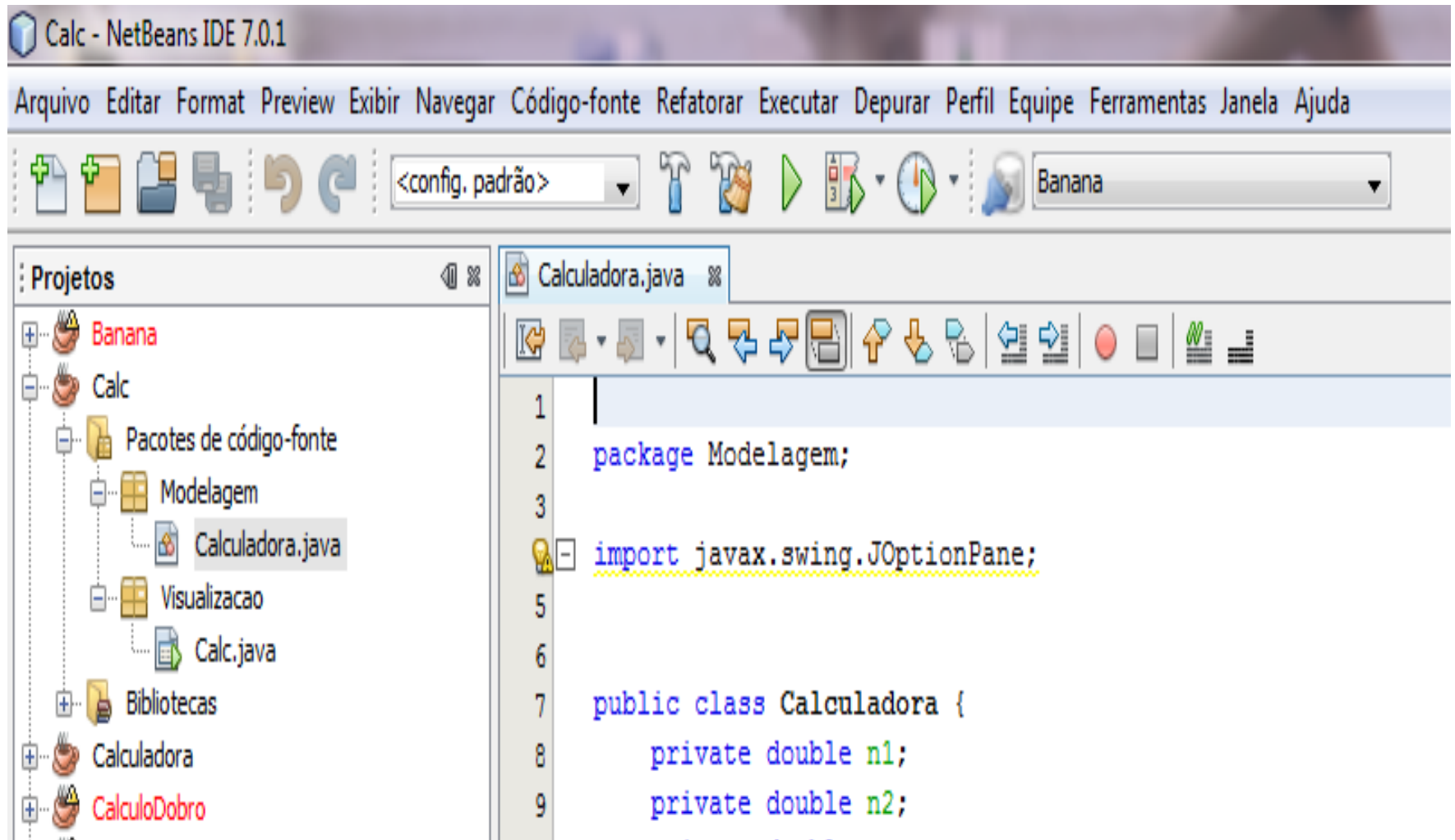
1. Clicar com o botão direito em cima do nome do projeto
2. Depois Novo -> Pacote Java
3. Em seguida o nome do pacote



## PARA CRIAR UMA CLASSE OU FORMULÁRIO

1. CLICAR COM O  
BOTÃO DIREITO NO  
NOME DO PROJETO





A janela do NetBeans

Criar os pacotes Controle, Modelagem,  
Visualização

Criar a classe no pacote Modelagem

Criar o formulário no pacote Visualização

Criar a classe de conexao no pacote

Controle

---

# Crie classe Conexao-pacote Controle

```
package Controle;
```

```
import java.sql.*;  
import javax.swing.JOptionPane;
```

```
public class Conexao {
```

```
    final private String driver = "com.mysql.jdbc.Driver";
```

```
    final private String url= "jdbc:mysql://127.0.0.1/banco";
```

```
    final private String usuario="root";
```

```
    final private String senha="";
```

```
    private Connection conexao;// objeto que faz conexao com o banco
```

```
    public Statement statement;// objeto que abre caminho até o banco, cria a autoestrada.
```

```
    public ResultSet resultSet;// objeto que armazena os comandos sql
```

```
    public boolean conecta() {
```

```
        boolean result = true;
```

```
        try {
```

```
            Class.forName(driver);
```

```
            conexao = DriverManager.getConnection(url,usuario,senha);
```

```
            //JOptionPane.showMessageDialog(null,"Conectou com o Banco de Dados");
```

```
        } catch(ClassNotFoundException Driver){
```

```
            JOptionPane.showMessageDialog(null,"Driver nao localizado: "+Driver);
```

```
            result = false;
```

```
        }catch(SQLException Fonte) {
```

```
            JOptionPane.showMessageDialog(null,"Erro na conexão com a fonte de dados: "+Fonte);
```

```
            result = false;
```

```
        }
```

```
        return result;
```

Nome do banco de dados

```
final private String driver = "com.mysql.jdbc.Driver";  
final private String url= "jdbc:mysql://127.0.0.1/banco";
```

Endereço do mysql

Pode usar alguns casos localhost

# Classe Conexao

```
public void desconecta () {
    boolean result = true;
    try
    {
        conexao.close();
        //JOptionPane.showMessageDialog(null, "Banco fechado");
    }
    catch(SQLException fecha)
    {
        JOptionPane.showMessageDialog(null, "Não foi possível fechar o banco de dados" + fecha);
        result = false;
    }
}

public void executeSQL(String sql) {
//chamada do metodo conecta para abrir a conexão com o db
    conecta();
    try{

        statement = conexao.createStatement();

        statement.execute(sql);
        //desconecta();
    }catch(SQLException sqle){
        JOptionPane.showMessageDialog(null, "Driver não encontrado!" + sqle.getMessage());
    }
}
```

# Classe Conexao

```
public ResultSet RetornarResultset(String sql){  
    ResultSet resultSet = null;  
    conecta();  
    try{  
        statement = conexao.createStatement();  
        resultSet = statement.executeQuery(sql);  
        resultSet.next();  
    }catch (Exception e){  
        JOptionPane.showMessageDialog(null, "Erro ao retornar resultset"+e.getMessage());  
    }  
    return resultSet;  
}  
  
}
```

# Criar a classe Cliente no pacote de Modelagem

```
package Modelagem;

import Controle.Conexao;
import java.sql.ResultSet;
import javax.swing.JOptionPane;

public class Cliente {
    private int codigo;
    private String nome;
    private String telefone;
    private String email;

    Conexao conCliente= new Conexao();
    public Cliente() {
        this(0, "", "", "");
    }
}
```

```
public Cliente(int codigo, String nome, String telefone, String email) {
    this.codigo = codigo;
    this.nome = nome;
    this.telefone = telefone;
    this.email = email;
}
```

## Cliente

-codigo: int  
-nome: String  
-telefone: String  
-email: String

+ gravar()  
+ consultar()



# Classe Cliente

```
public int getCodigo() {  
    return codigo;  
}  
public void setCodigo(int codigo) {  
    this.codigo = codigo;  
}  
public String getNome() {  
    return nome;  
}  
public void setNome(String nome) {  
    this.nome = nome;  
}  
public String getTelefone() {  
    return telefone;  
}  
public void setTelefone(String telefone) {  
    this.telefone = telefone;  
}  
public String getEmail() {  
    return email;  
}  
public void setEmail(String email) {  
    this.email = email;  
}
```

# Classe Cliente

```
public void gravar() {  
    String sql;  
    sql="Insert into Cliente(Codigo,Nome,telefone,Email)values"+  
    "("+this.getCodigo()+", '"+this.getNome()+"', '"+this.getTelefone()+"', '"+this.getEmail()+"'";  
    conCliente.executeUpdate(sql);  
    JOptionPane.showMessageDialog(null,"Registro gravado com sucesso...");  
}  
  
public ResultSet consultar() {  
    ResultSet tabela;  
    tabela = null;  
  
    String sql= "Select * from cliente";  
    tabela = conCliente.RetornarResultset(sql);  
    return tabela;  
}
```

# Código do formulario

Cadastrar Cliente

Codigo

Nome

Telefone

Email

Codigo	Nome	Email
1	Andreia	andreia@hotmail.com
2	Jose	abndmd@hotmail
3	Simone	simone@hotmail
4	Gilda	gilda@hotmail

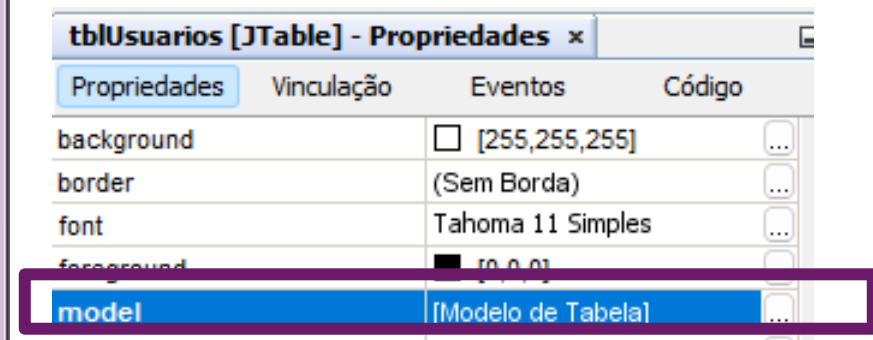
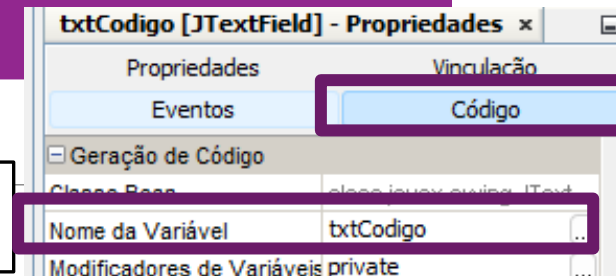
Componente JTextField /campo de texto  
Nome da variável: txtCodigo

Componente JTextField /campo de texto  
Nome da variável: txtNome

Componente JTextField /campo de texto  
Nome da variável: txtEmail

Componente JButton  
Nome da variável: btnIncluir

Componente Jtabela / Tabela  
Nome da variável: tblUsuarios



Para adicionar as colunas na tabela altere a propriedade **model** e defina os títulos das colunas

# Código do formulario

```
package Visualização;
```

```
import Modelagem.Cliente;
```

```
import java.sql.ResultSet;
```

```
import java.sql.SQLException;
```

```
import javax.swing.JOptionPane;
```

```
import javax.swing.table.DefaultTableModel;
```

```
public class FCliente extends javax.swing.JFrame {
```

```
    public FCliente() {
```

```
        initComponents();
```

```
    }
```

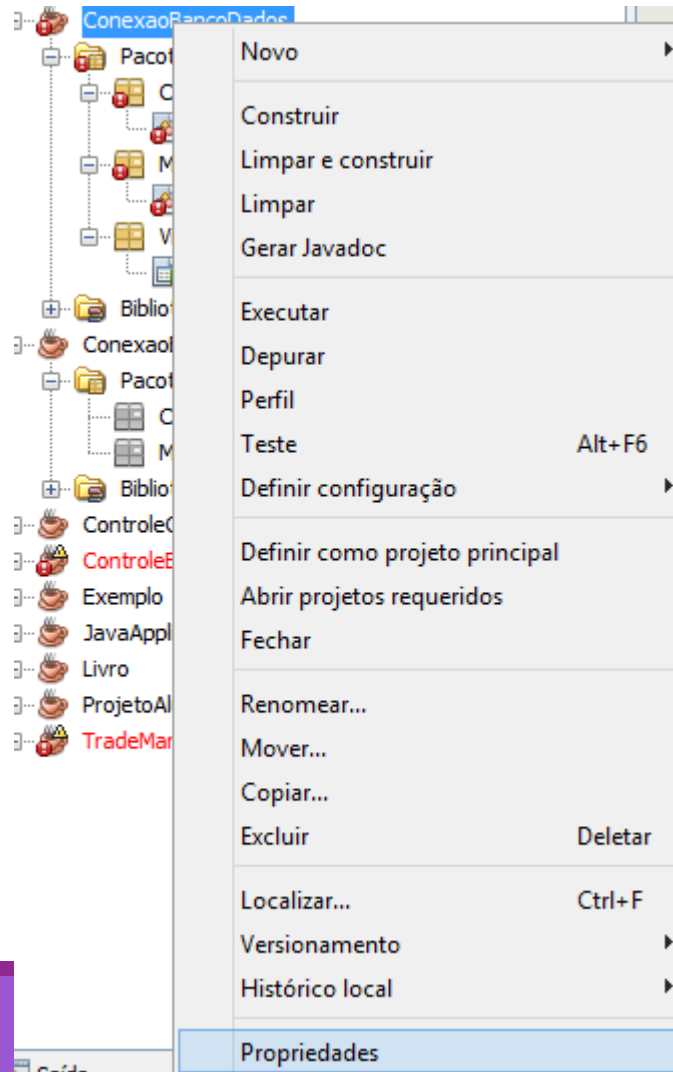
```
    Cliente cli = new Cliente();
```

# Código do formulário – Incluir

```
private void btnIncluirActionPerformed(java.awt.event.ActionEvent evt) {  
    cli.setCodigo(Integer.parseInt(txtCodigo.getText()));  
    cli.setNome(txtnome.getText());  
    cli.setEndereco(txtendereco.getText());  
    cli.setTelefone(txttelefone.getText());  
    cli.gravar();  
  
    ResultSet tabela;  
    tabela = null;  
  
    tabela = cli.consultar();  
    DefaultTableModel modelo = (DefaultTableModel) jTable2.getModel();  
    modelo.setNumRows(0);  
    try  
    {  
        do{  
            modelo.addRow(new String[]{tabela.getString(1), tabela.getString(2), tabela.getString(4)});  
        }  
        while(tabela.next());  
    }catch(SQLException erro)  
    {  
        JOptionPane.showMessageDialog(null, "Erro ao preencher tabela"+ erro) ;  
    }  
}
```

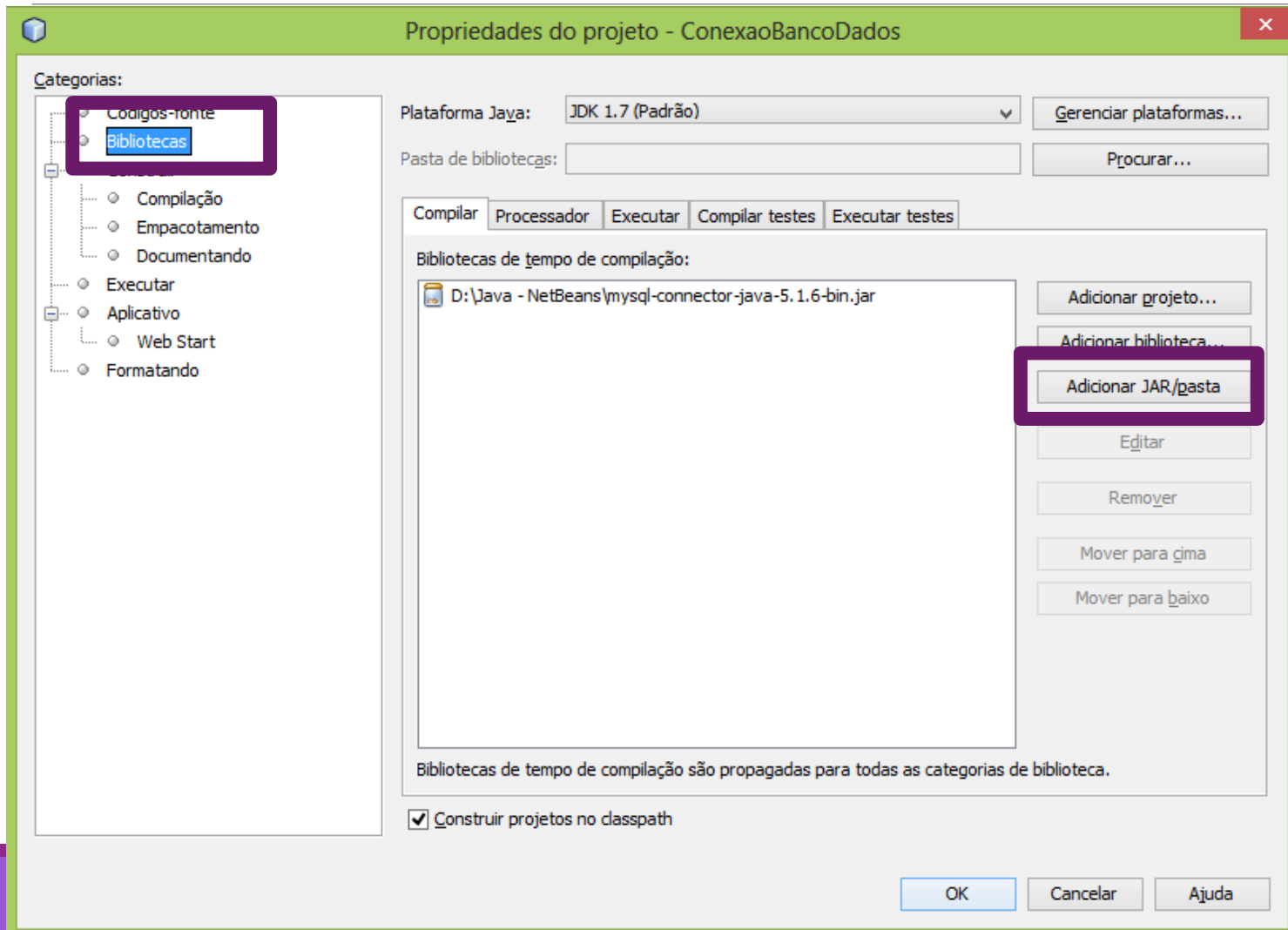
# Incluir Conector -MySQL

Clique com botão direito nome do projeto ->Propriedades



# Incluir Conector -MySQL

Clique em bibliotecas , em seguida no botão Adicionar JAR e adicione o conector mysql



# Crie a tabela no BANCO DE DADOS

No WorkBench , crie um database com nome banco, e crie a tabela Cliente

```
create database banco;  
use banco;  
create table cliente(  
id int(4) auto_increment primary key,  
codigo int(4),  
Nome varchar(60) ,  
Endereco varchar(100),  
Telefone varchar(18));
```



# Exercicio 1

1. Crie os pacotes de Modelagem, Controle e Visualização.
2. Crie a classe Usuario em seu pacote designado, com os atributos nome, login, senha, email.
3. Crie o banco de dados chamado Usuario Com os seguintes campos nome, login, senha, email
4. Crie a classe de Conexão.
5. Crie a formulário FUsuario.

The screenshot shows a Java Swing window titled "Cadastro Usuário". The window has a light gray background and a title bar with standard Windows controls. The main content area is titled "Cadastro de Usuário" in a bold, red, italicized font. Below the title, there are four labels with corresponding text input fields: "Digite o Nome:", "Digite o e-mail:", "Digite o login:", and "Digite a senha:". Each label is in a bold, italicized font. Below the input fields, there are three buttons: "Cadastrar" (orange), "Limpar" (pink), and "Sair" (orange). At the bottom of the window, there is a table with four columns: "Nome", "Login", "Email", and "Senha". The table has five rows, with the first row being the header and the subsequent four rows being empty data rows.

Nome	Login	Email	Senha

## Usuario

-nome: String  
-email: String  
-login: String  
-senha: String

+cadastrarUsuario()  
+listarUsuario()

# Exercicio 2

1. Crie os pacotes de Modelagem, Controle e Visualização.
2. Crie a classe Produto em seu pacote designado, com os atributos codigo, nomeProduto e Descrição.
3. Crie o banco de dados chamado ConsultaProduto. Com os seguintes campos: codigo, nomeProduto e Descrição.
4. Crie a classe de Conexão.
5. Crie a formulário FProduto.

*Cadastro de Produto*

Codigo

Nome do Produto

Descrição

Codigo	Nome do Produto	Descrição

## Produto

-codigo: int  
-nomeProduto: String  
-descricao: String

+cadastrarProduto()  
+listarProduto()

# Exercicio 3

1. Crie os pacotes de Modelagem, Controle e Visualização.
2. Crie a classe Venda em seu pacote designado, com os atributos codigo, nomeVendedor e NomeProduto.
3. Crie o banco de dados chamado ConsultaVenda. Com os seguintes campos: codigo, nomeVendedor e NomeProduto.
4. Crie a classe de Conexão.
5. Crie a formulário FVenda.

*Cadastro de Venda*

Codigo

Nome do Vendedor

Nome do produto

Codigo	Nome do Vendedor	Nome do produto

Gravar Limpar sair

## Venda

-codigo: int  
-nomeVendedor: String  
-nomeProduto: String

+cadastrarVenda()  
+listarVenda()